Mobile Payment Protocol for Tag-to-Tag Near Field Communication (NFC)

http://dx.doi.org/10.3991/ijim.v6i4.2166

Emir Husni, Kuspriyanto, Noor Basjaruddin
Institut Teknologi Bandung, Bandung, Indonesia

***Abstract*—Communication between the near field communication (NFC) devices occurs in a very close distance of less than 10 cm. In the NFC-based payment system, close proximity between devices will increase the security of transactions. The disadvantage is the interaction between devices requires more physical activity of device owners because the device must be brought near to other devices some times. Besides requiring more physical activity, NFC-based interaction also takes a longer time because the device needs to be moved from one position to another. This paper proposed secure and efficient protocol that will reduce the physical activity of the device owners and reducing transaction time. The data exchange between merchant and payer will be executed without waiting for each other and one transaction will require two data transmissions are performed by the merchant and payer. Transactions are secured by the use of encryption on each data which sent by the merchant and payer. In addition, the protocol also guarantees the security of offline micro transactions and online macros transactions.**

***Index Terms*—NFC-based payment, transaction protocol, mobile payment, secure protocol.**

I. INTRODUCTION

Near field communication (NFC) is a short-range wireless technology which is the advanced development of RFID technology. NFC's fundamental advantages compared to other wireless technologies like Bluetooth is the availability of the data storage facility known as the NFC tag. NFC is not just a replacement data cable as Bluetooth, but also as a means of store of data. Referring to the NFC Forum, NFC technology is currently used in three areas, namely sharing, pairing, and transaction. The three areas which are the use of NFC were developed with full support from various handset vendors.

Wireless communication between the NFC devices, the NFC tag and the availability of NFC handsets support the development of contactless payment and the actual e-wallet [1].

A protocol is needed to regulate the communication between two NFC devices. In payment system two devices are payer's device and merchant's device. The process of communication between NFC devices implemented with the use of read / write tag facility, so it requires a protocol called protocol tag to tag. Writing or reading of the tag carried by each bring two NFC devices and is known as a tap.

Protocols that are currently available is the protocol that manages the communication between the NFC with a peer to peer communication mode. This protocol needs a sequential process during data exchange. The mutual wait in the transaction process will require a longer time.

This paper introduces the tag to tag protocol that is used in the macro and micro payments. This protocol is different from the existing protocols, especially in the process of data exchange that does not require waiting for each other and which party should initiate the data exchange.

II. PROTOCOL DESIGN

A. Micropayment and Macropayment

Micropayment is a transaction amount between USD 10 cent - USD 10. This payment can occur between merchant and payer or between NFC account owners. Micropayment is off-line transaction as e-wallet. Macropayment is a transaction of more than USD 10 to USD 200. Macropayment is online transaction in order to improve transaction security [2-5].

B. Protocol specific assumptions and notations

Protocol is designed for communication between merchant and payer. The assumptions used in this protocol is the amount to be paid by the payer is known by merchant from NFC tags.

The working principle of the protocol tag to tag is set up communication between NFC devices which carried out the data exchange process without which the device is determined to start communication. The second NFC device can initiate communication and mutual prosenya no waiting.

The principle is not in communication with each other waiting to tag the selected tag in order for the communication process more efficient in terms of time.

1) Notations:

{P, M, TP}:
a set of engaging parties payer, merchant, and third party.
X->Y:
X sends message to Y.
{ X || Y }:
a set of messages or message components.
E_K [X]:
X symmetrically encrypted with the key K.
H(.):
one-way hash function such as MD5
ID_X:
the identity (ID) of party X.
TID:
ID of the transaction. It is chosen by merchant and uniquely identifies a transaction.

{ $ACCNO_P$, $ACCNO_M$ }:
a set of payment account numbers of the payer and the merchant.
AMOUNT:
amount of the transaction.
PSWD:
payment password of the payer.
ACCBALANCE:
Account balance of the payer.
$IDENT_X$:
Identification data of party x. This data more complete than ID.
PAY_X:
Pay status of party X. It is one of the values:ACCEPT, REJECT
K_i:
Every user produces similar K_i using the same secret formula. Variables of the formula are predetermined and one of them given periodically by the third party. This can be used for banning accounts.
K_{SM}:
Secret session key generated by merchant. It is at least 128bits for security.
K_{SP}:
Secret session key generated by payer. It is at least 128bits for security.
MAC_X:
The message authentication codes from party X.
$RAND_X$:
The random number from party X.

C. Micropayment Protocol

Micropayment protocol is shown in Fig. 1.

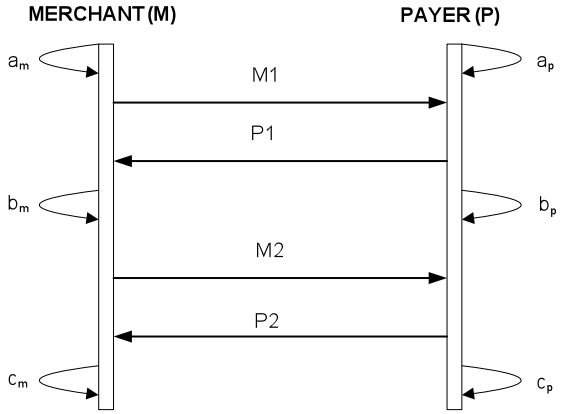

Figure 1. Micropayment protocol.

(a_m, a_p) M: P:

K_{SM}= E_{Ki} ($IDENT_M$)
K_{SP}= E_{Ki} ($IDENT_P$)

Merchant and payer generate shared session keys (K_{SM} and K_{SP}) which is used for encryption and decryption.

(M1) M->P:

$mm1=ID_M$ || TID || AMOUNT || E_{Ki}[$H(ACCNO_M)$] || MAC_M || K_{SM};

The merchant sends a message to payer which containing merchant's ID number, transaction number, payment amount, encryption of merchant's account number, MAC_M, and K_{SM}.

(P1) P->M:

$pm1=ID_P$ || TID || ACCBALANCE || E_{Ki}[$H(ACCNO_P$ || PSWD)] || MAC_P || K_{SP};

The payer sends a message to merchant which containing payer's ID number, transaction number, account balance, encryption of payer's account number and password, MAC_P, and K_{SP}.

(b_m) M:

Merchant check payment amount and payer's account balance. If AMOUNT < ACCBALANCE then value of PAY_M is ACCEPT. Otherwise if AMOUNT > ACCBALANCE then the value of PAY_M is REJECT.

(b_p) P:

Payer check payment amount and account balance. If AMOUNT < ACCBALANCE then value of PAY_P is ACCEPT. Otherwise if AMOUNT > ACCBALANCE then the value of PAY_P is REJECT.

The status of PAY_P is validated by confirmation of the payer by pressing the 'OK' or 'NO' on screen.

(M2) M->P:

$mm2=ID_M$ || TID || AMOUNT || PAY_M || E_{Ki}[$H(ACCNO_M$)] || MAC_M || K_{SM};

The merchant sends a message to payer which containing merchant's ID number, transaction number, payment amount, PAY_M, encryption of merchant's account number, MAC_M, and K_{SM}.

(P2) P->M:

$pm2=ID_P$ || TID || ACCBALANCE || PAY_P || E_{Ki}[$H(ACCNO_P$ ||PSWD)] || MAC_P || K_{SP};

The payer sends a message to merchant which containing payer's ID number, transaction number, account balance, PAY_P, encryption of payer's account number and password, MAC_P, and K_{SP}.

(c_m) M:

The merchant verifies the value of PAY_P. If PAY_P ='ACCEPT'&PAY_M='ACCEPT' then the payer's payment fund is deposited in a merchant account and a purchase receipt is printed.

(c_p) P:

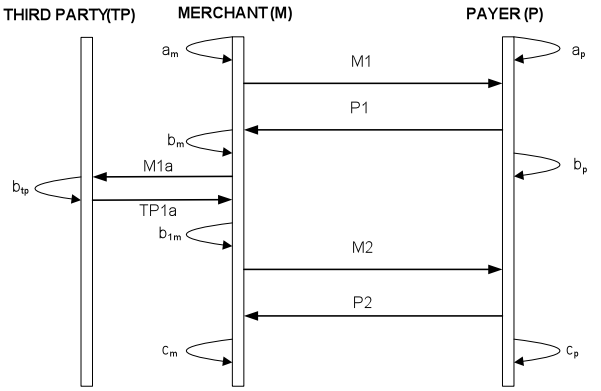

Figure 2. Macropayment protocol.

The payer verifies the value of PAY_M. If PAY_P ='ACCEPT'&PAY_M='ACCEPT' then the payer's accout balance is reduced and a payment status is displayed on screen.

D. Macropayment Protocol

Macropayment protocol is shown in Fig. 2.

(a_m, a_p) M, P:

K_{SM}= E_{Ki} ($IDENT_M$)
K_{SP}= E_{Ki} ($IDENT_P$)

Merchant and payer generate shared session keys (K_{SM} and K_{SP}) which is used for encryption and decryption.

(M1) M->P:

$mm1=ID_M$ || TID || AMOUNT || E_{Ki}[$H(ACCNO_M)$] || MAC_M || K_{SM};

The merchant sends a message to payer which containing merchant's ID number, transaction number, payment amount, encryption of merchant's account number, MAC_M, and K_{SM}.

(P1) P->M:

$pm1=ID_P$ || TID || ACCBALANCE || E_{Ki}[$H(ACCNO_P$ || PSWD)] || MAC_P || K_{SP};

The payer sends a message to merchant which containing payer's ID number, transaction number, account balance, encryption of payer's account number and password, MAC_P, and K_{SP}.

(b_m) M:

Merchant check payment amount and payer's account balance. If AMOUNT < ACCBALANCE then value of PAY_M is ACCEPT. Otherwise if AMOUNT > ACCBALANCE then the value of PAY_M is REJECT.

(M1a) M->TP

$mm1a=ID_M$ || ID_P || TID || AMOUNT || TIMESTAMP || E_{Ki}[$H(ACCNO_M)$] || MAC_M || K_{SM};

The merchant sends a message to third party which containing merchant's ID number, payer's ID number, transaction number, payment amount, timestamp, encryption of merchant's account number, MAC_M, and K_{SM}.

(TP1a) TP->M

$tpm=ID_M$ || ID_P || TID || AMOUNT || TIMESTAMP || E_{Ki}[$H(ACCNO_{TP})$] || MAC_{TP} || $RAND_{TP}$

The third party sends a message to merchant which containing merchant's ID number, payer's ID number, transaction number, payment amount, timestamp, encryption of third party's account number, MAC_{TP}, and K_{TP}, and random number.

(b_{1m}) M:

The merchant checks random number from TP ($RAND_{TP}$). If the random number is TRUE and PAY_M is ACCEPT then PAY_M is validated. If random number is FALSE then PAY_M is REJECT.

(b_p) P:

Payer check payment amount and account balance. If AMOUNT < ACCBALANCE then value of PAY_P is ACCEPT. Otherwise if AMOUNT > ACCBAANCE then the value of PAY_P is REJECT.

The status of PAY_P is validated by confirmation of the payer by pressing the 'OK' or 'NO' on screen.

(M2) M->P:

$mm2=ID_M$ || TID || AMOUNT || PAY_M || E_{Ki}[$H(ACCNO_M$)] || MAC_M || K_{SM};

The merchant sends a message to payer which containing merchant's ID number, transaction number, payment amount, PAY_M, encryption of merchant's account number, MAC_M, and K_{SM}.

(P2) P->M:

$pm2=ID_P$ || TID || ACCBALANCE || PAY_P || E_{Ki}[$H(ACCNO_P$ ||PSWD)] || MAC_P || K_{SP};

The payer sends a message to merchant which containing payer's ID number, transaction number, account balance, PAY_P, encryption of payer's account number and password, MAC_P, and K_{SP}.

(c_m) M:

The merchant verifies the value of PAY_P. If PAY_P ='ACCEPT'&PAY_M='ACCEPT' then the payer's payment fund is deposited in a merchant account and a purchase receipt is printed.

(c_p) P:

The payer verifies the value of PAY_M. If PAY_P ='ACCEPT'&PAY_M='ACCEPT' then the payer's accout balance is reduced and a payment status is displayed on screen.

III. IMPLEMENTATION

Fig. 3 shows implementation of tag to tag protocol in micropayment. NFC's user or account owner chooses product items to be purchased by way of reading the product tag. After selecting the product, NFC user will verify the purchase and the amount to be paid by transferring data on NFC handset to the NFC payment system. Monitor display will show the results of verification. Once product items and the amount to be paid are correct, user will pay through the NFC payments application. The receipt will be used by the inspector to check product items. Macropayment process is shown in Fig. 4. The difference between macropayment and micropyment is when user will pay. In macropayment, before the user pays the merchant asked for verification from the third party. Payment process can be done if the third party permits.

The applications were developed using Android 2.3 operating system smartphones which are Samsung Nexus S. All applications were programmed using Android software development kits (SDK). The efficient tag-to-tag NFC protocol system proposed in this paper has been developed and run well. Process payments using NFC can be seen in Figure 5.

IV. SECURITY ANALYSIS

In this paper K_i is important key which is generated using the formula $K_i = f_{ki}$ (a, b, c, ...) where a, b, c, … are secret variables. Every user produces similar K_i where variables of the formula are predetermined and one of them given periodically by the third party. This key can be used for banning accounts. If two users do not have similar K_is, the transaction cannot be continued.

A. Efficiency

All encryption and decryption protocols computed in mobile devices are symmetric encryption, so the computations in mobile devices need only a small resource. In micropayment protocol, it only computes encryption and decryption twice in mobile device. In the macropayment protocol, the encryption and decryption protocols computed three times. Other computations which need more resources like asymmetric encryption and decryption are processed in a server. Moreover, the tag-to-tag protocol has better time efficiency than ordinary protocol because it does two processes in every unit of time.

B. Confidentiality

In the protocol, the sensitive part of payment information, such as $ACCNO_P$, $ACCNO_M$, $ACCNO_{TP}$ and PSWD, are encrypted by the secret key which are only shared by the payer and the merchant or by the payer and the third party. When merchant receives a payment message from payer, PAY_P (unencrypted part of the message) is verified by the merchant to know the status of payer's payment (rejected or not). The payer's session key is used for decrypting the ciphertext part of the message. Therefore, the payer's privacy of payment transaction is preserved because the sensitive part of the message is encrypted. This encrypted part cannot decrypted by anyone not having the session key.

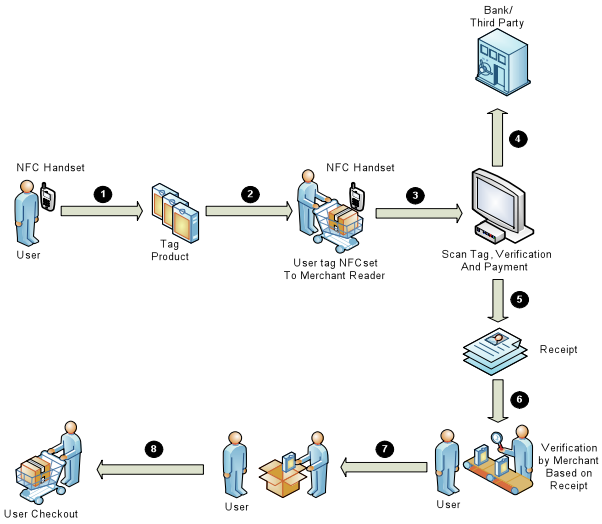

Figure 3. Micropayment process.

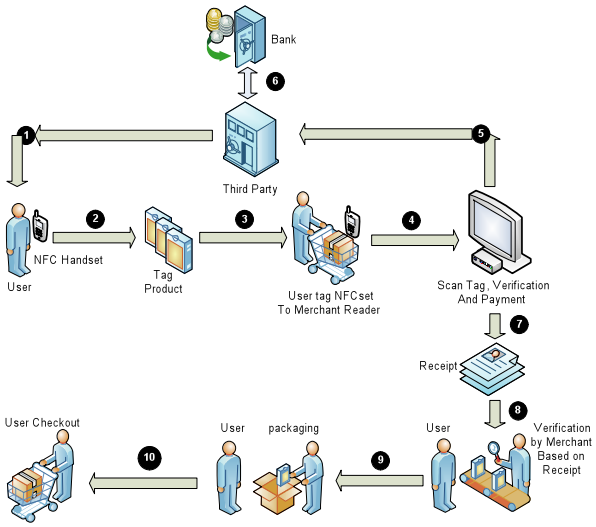

Figure 4. Macropayment process.

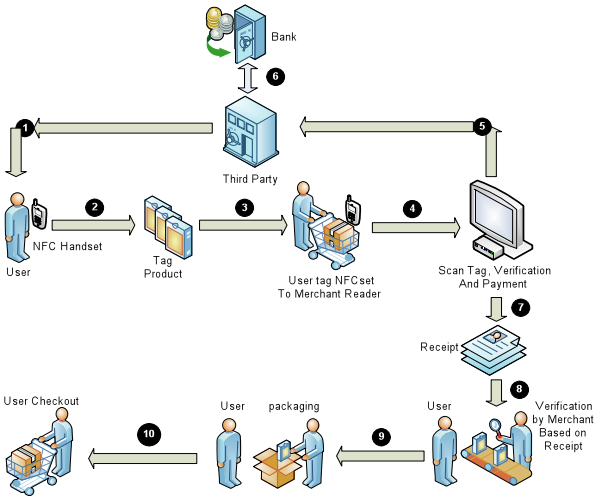

Figure 5. Payment process.

C. Difficulty of alteration

The protocol allows the issuer to detect whether the significant data have been altered. Suppose that a dishonest person wants to modify the data in the transaction, he replaces the original value of data with a different value data*. In this case he should also replace the value of the message authentication codes (MAC). In the protocol, no person other than the final receiver knows the secret string. Consequently other persons cannot directly compute the MAC of the fake message. Since the hash function is computation resistant, then it is computationally infeasible for the other person to compute the MAC for any value data* different from the original value data. When the dishonest person sends data* to the receiver, the receiver will compute the MAC and find out that this value is different from the one stored in the message. Therefore, the receiver will reject the message.

D. Difficulty of alteration

The protocol allows the issuer to detect whether the significant data have been altered. Suppose that a dishonest person wants to modify the data in the transaction, he replaces the original value of data with a different value data*. In this case he should also replace the value of the message authentication codes (MAC). In the protocol, no person other than the final receiver knows the secret string. Consequently other persons cannot directly compute the MAC of the fake message. Since the hash function is computation resistant, then it is computationally infeasible for the other person to compute the MAC for any value data* different from the original value data. When the dishonest person sends data* to the receiver, the receiver will compute the MAC and find out that this value is different from the one stored in the message. Therefore, the receiver will reject the message.

E. Fund security

The protocol takes effective measures of fund allocation to protect the benefits of both merchants and payers. After merchant receiving the payer's authorization, then the payer's payment fund is deposited in a merchant account. On the one hand, the payer gives the merchant an assurance that the payment fund will be used for the transaction and the merchant can go on business with the payer. On the other hand, once there is not enough fund in payer account then the transaction automatically discard and avoid the payer loss money.

V. CONCLUSION

This paper explained an effective tag to tag NFC Protocol for secure mobile payment using mobile devices. The protocol has the advantages as follows: (1) The protocol does not require too much computational resource for optimizing the payment processes. (2) The off-line-update key mechanism enhances the security of mobile payment. (3) Sensitive payment data is encrypted by one-session-one-key so that the protocol preserves payers' privacy. (4) The protocol takes effective measures of fund allocation to protect the benefits of both merchants and payers.

REFERENCES

[1] H.C. Cheng, J.W. Chen, T.Y. Chi, P.H. Chen, "A Generic Model for NFC-based Mobile Commerce," 11th ICACT 2009, pp. 2009-2014, 2009.

[2] L. Xi & H.H. Ping, "Efficient Protocol of Secure Mobile Payment," Journal of Communication and Computer, Vol.4 No.5 (Serial No.30), May 2007.

[3] T.S. Fun, L.Y. Beng, R. Roslan, and H.S. Habeeb, "Privacy in New Mobile Payment Protocol," World Academy of Science, Engineering and Technology, 2008.

[4] M.N. Abdullah & M.T. Hadi, "A Secure Mobile Banking Using Kerberos Protocol," Eng.& Tech. Journal, Vol. 27, N0. 6, 2009.

[5] K. Chikomo, M.K. Chong, A. Arnab, and A. Hutchison, "Security of Mobile Banking," Data Networks Architecture, 2006.

AUTHORS

Emir Husni, Kuspriyanto, and Noor Basjaruddin are with School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Jl. Ganesha no. 10, Bandung, Indonesia.

Received 26 June 2012. Published as resubmitted by the authors 23 September 2012.